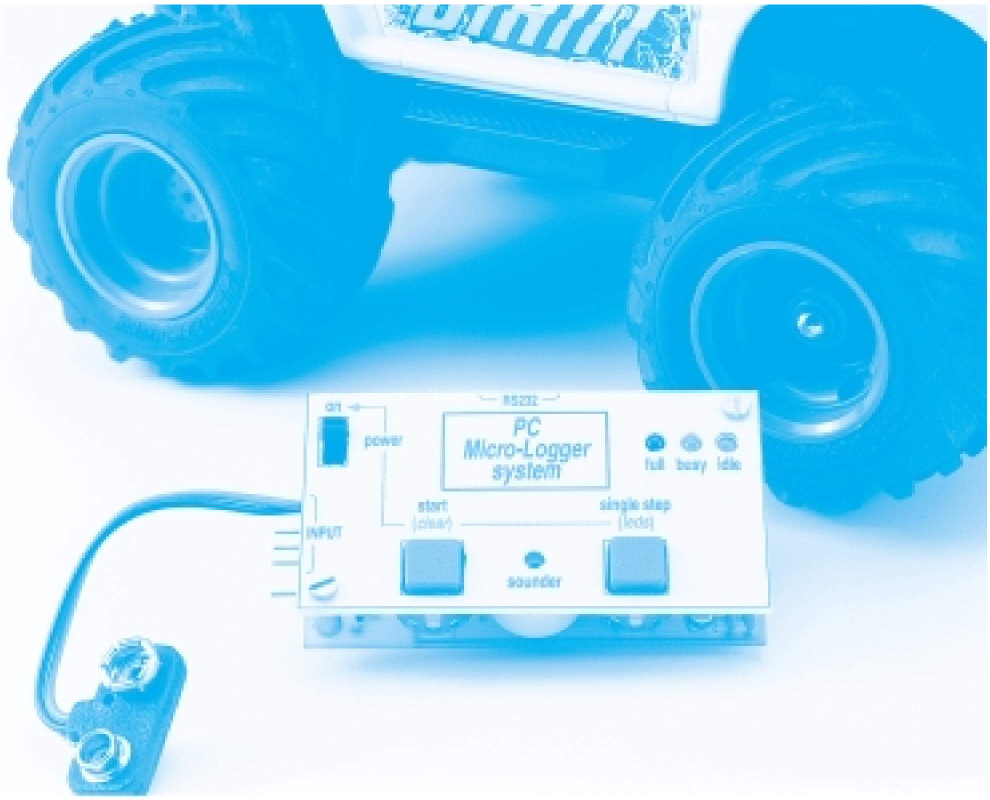
The micrologger presented here is remarkable because this device, no bigger than a matchbox, is a data logger and voltmeter that can operate fully on its own. In addition, it has an RS232 port for connection to the COM port of a PC. Commands to be executed by the micrologger's built-in command processor can be entered via the keyboard of the PC, and the measurement result is displayed on the screen of the monitor. After being configured via the PC, the micrologger can be disconnected and used stand-alone. It can store up to 82 10-bit measurement values in EEPROM. Stored measurements are saved even when the power is switched off. After the micrologger has again been connected to the PC, the measurement data can be uploaded and converted into a graphic display, using a spreadsheet program such as KyPlot.



Design by B. Stuurman

micro datalogger

for use with a PC and stand-alone

Micrologger specifications

- operates as a standalone data logger and 'single-stepper' with data postprocessing on a PC, as well as a PC voltmeter/datalogger and 'single-stepper'
- non-volatile memory for 82 3-digit samples
- sampling delay adjustable from 1 second to 15:59:59 h:m:s
- maximum measurement session duration approximately 1300 hours
- RS232 interface; protocol 38400 (19200), 7, n, 2
- screen update rate 3x per second
- range / basic accuracy

range	basic accuracy
0 to 2.49 V	±(0.5%, 1 d)
2.5 to 9.99 V	±(0.5%, 3 d)
10.0 to 99.9 V	±(0.5%, 2 d)

- input impedance: approx. 500 kΩ
- power supply: 9-V battery
- current consumption: 4.6 mA (LEDs on); 2.8 mA (LEDs off)
- dimensions (exclusive of protruding connector pins): 64×36×15 mm (l × w × h)
- weight without battery: approx. 30 g

Its small dimensions and stand-alone operation capability make the micrologger an outstanding choice for 'remote' measurements. For example, it could be used to measure the voltage and current consumption of the motor of an electrically driven model, to allow the drive components to be optimized. Amateur meteorologists could collect measurements over a long period of time without having to be personally present. Two data loggers could be carried aloft by a weather balloon, one to measure the temperature and the other to measure the altitude. In physics exercises in which several teams take measurements, each team could be provided with a micrologger to store their data. The advantage of this approach is that an expensive collection of instruments is not needed to perform the measurements, and at the same time all measurements are registered by team and can be subsequently processed.

When the micrologger is connected to a PC, the command 'volt' causes the value of the actual voltage to be sent to the PC three times per second, via the RS232 interface. The PC screen acts as the display, and if a 'full-screen' character size is used, the display can be read at a distance of several metres.

An introduction

Figure 1 shows the complete schematic of the micrologger. At its heart is an ST62T60, a powerful 8-bit microcontroller from SGS-Thomson. The following peripheral components are connected to the core of the controller via an internal serial data bus:

- a normal timer,
- a timer with a pulse-width modulated (PWM) output,
- a watchdog timer,
- an 8-bit A/D converter,
- a serial interface (Serial Peripheral Interface, or SPI),
- 128 bytes of nonvolatile memory (EEPROM).

The normal timer is always active and looks after the timing of several different software processes. This takes place in the interrupt routine by means of timer flags. Since this timer is also used to generate the seconds flag, it is always loaded with a fixed value that yields a 'nearly perfect' seconds clock after some further division.

The PWM timer is used to generate a pulse waveform signal (on the ARTIM output). The average value of this signal is used to increase the resolution of the 8-bit A/D converter to 10 bits. We'll say more about this shortly.

The watchdog timer is used for serial I/O timing, to free up the other timers. This is possible since the watchdog timer can be activated under software control. Except for this timer and the SPI, no hardware is used for the basic I/O function.

The 8-bit A/D converter is naturally used for voltage measurements. To be precise, the voltage on the analogue input (Ain) is measured 64 times for each sample, and the result is then divided by 64. The range of the A/D converter is from 00_H to FF_H. The first of these values will be measured if the voltage on Ain is equal to the voltage on the Vss pin (0V), and the second value will be measured if the voltage on Ain is equal to voltage on the Vdd pin (+5V). The accuracy of the micrologger is thus directly related to the accuracy of the supply voltage (Vdd). For this reason, an LP2950CZ5.0 has been chosen for the voltage regulator. Not only is this a low-drop regulator, it is also recommended as a reference source by its manufacturer!

Current sourcing or sinking by other components in the circuit can affect the conversion, due to voltage drops in the wiring. For this reason, the LEDs are switched off while a measurement is being made. This produces a characteristic blinking effect.

The serial interface serves for communication with a PC. It allows commands to be sent to the micrologger, and data to be sent from the micrologger to the PC. The micrologger contains a true interactive command processor. Of course, a terminal emulator program, such as HyperTerminal (supplied with Windows), must be present on the PC.

The 128-byte EEPROM is used to store various settings, including the sampling delay parameter, and it also includes room for 82 11-bit sample values (10 data bits and a decimal point). The samples are written in an encoded form, with two samples stored in every three bytes of memory to make the best use of the available capacity.

More details

The RS232 interface

The ability to communicate with a PC via the RS232 interface is vitally important for this application. However, this creates

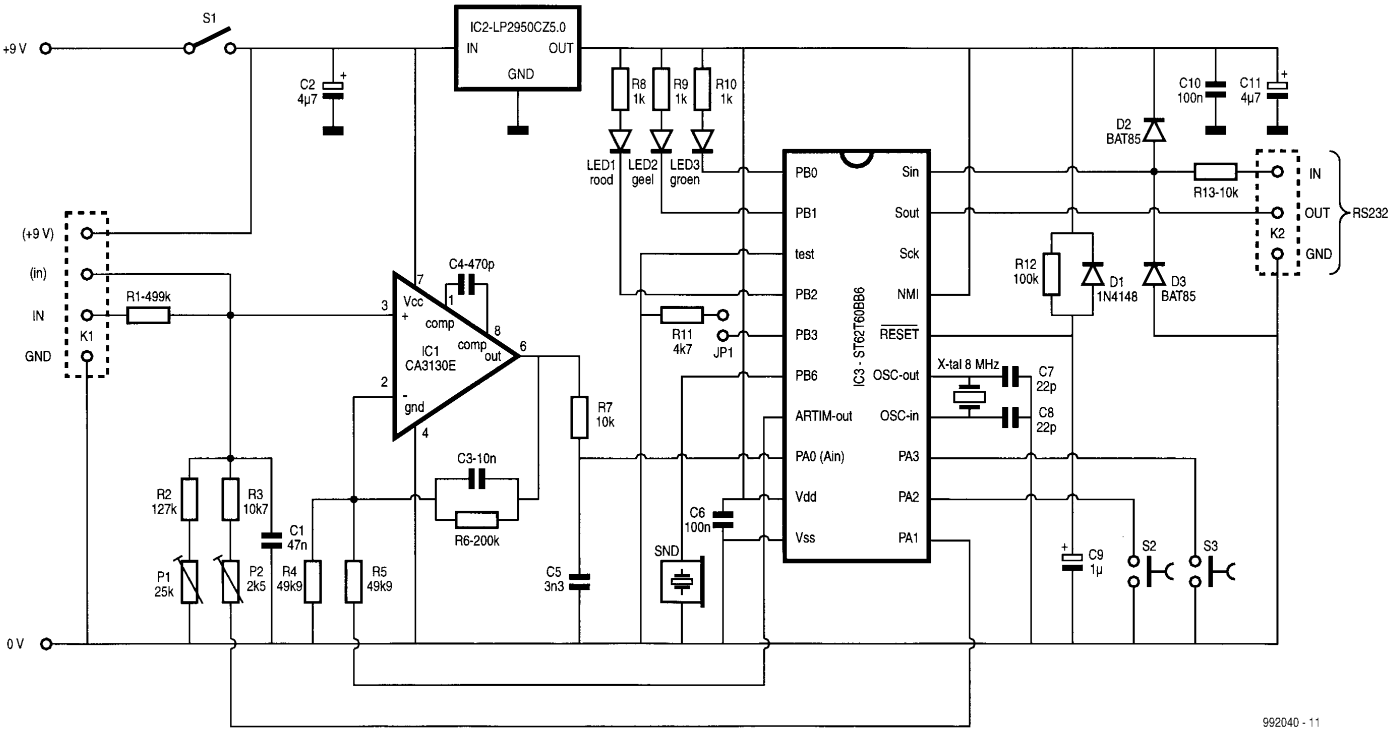


Figure 1. A microcontroller, an opamp and a handful of components are all you need to build a mini PC data logger and voltmeter.

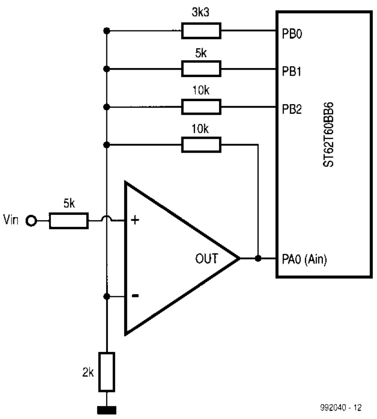


Figure 2. Using an opamp as an adder amplifier to increase the resolution from 8 to 10 bits.

a problem, since this communication must not be allowed to disturb the normal operation of the micrologger.

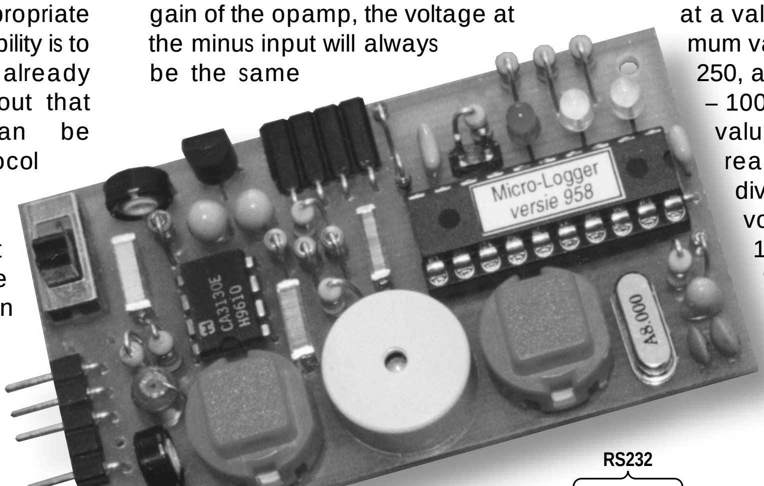
Usually, an RS232 interface is built up using software, in which the baud rate is determined by delay loops, or with the aid of timers. However, here the timers are indispensable for other purposes, and the software approach is only possible for relatively low baud rates, which would not be appropriate here. The only remaining possibility is to use the SPI, which after all is already present. Ultimately, it turns out that RS232 communication can be achieved by using a 7-bit protocol and look-up tables. During normal operation, the SPI is set up to receive data. A start bit at Sin (see **Figure 1**) triggers the SPI, which subsequently takes in the data all on its own. On completion, the SPI generates an interrupt, which is used to set a flag. This flag is regularly polled in the main program loop, which can take the necessary actions if it is set. The hardware for the RS232 interface is minimal — two Schottky diodes (for speed) and one resistor. The only thing that is necessary is that the connection cable includes a number of cross-connections that serve to 'trick' the UART of the PC (see the January 1999 issue of Elektor for more information). The maximum speed of the SPI is very high, but due to the clock rate of 8MHz and the available predivider settings, only 38400 baud and 19200 baud are actually possible. The latter value can be selected by installing jumper JP1.

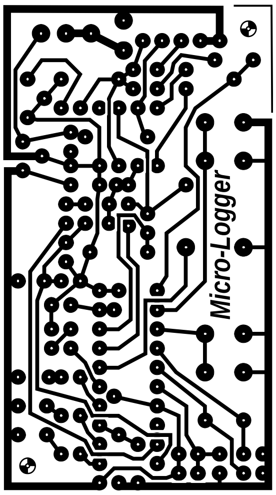
From 8-bit to 10-bit ADC resolution

Although the resolution of the A/D converter of the ST62T60 is 8 bits, it is possible to increase this resolution with an opamp. The basic idea, illustrated in **Figure 2**, comes from an ST application handbook [1].

The opamp is wired as a noninverting amplifier with negative feedback. Due to the negative feedback and the high gain of the opamp, the voltage at the minus input will always be the same as the voltage at the plus input. When Vin reaches a level that causes the A/D converter to produce its maximum output value, PB2 is set high. The resistor values are chosen so that the output voltage of the opamp then goes to zero, which allows the voltage on Vin to continue to increase until the A/D converter again reaches its maximum output value. When this happens, PB1 is set high and the opamp's output voltage again goes to zero. The measurement range of the A/D converter is thus used four times over, so that the resolution is increased from 8 to 10 bits.

Now let's return to **Figure 1**. The CA3130 opamp has two special properties, which are that its inputs can handle voltages down to just below earth level and that the output voltage range extends down to the earth level. This opamp is connected directly to the +9V supply, so that its output has no difficulty reaching 5V, which is the maximum input level of the A/D converter. The PWM signal (ARTIM out) is connected to the minus input of the opamp via R5. There are four working ranges: 0%, 25%, 50% and 100%. To simplify the calculations, the switching points have been chosen to occur at a value of 251 (FB_H). The maximum value for each range is thus 250, and the total range is thus 0 – 1000. The software tests for a value of 1000 and, if this is reached, switches in a 10x divider by setting PA1 low. For voltage readings under 1000, P1 is configured as a 'floating' input. The voltmeter thus has two





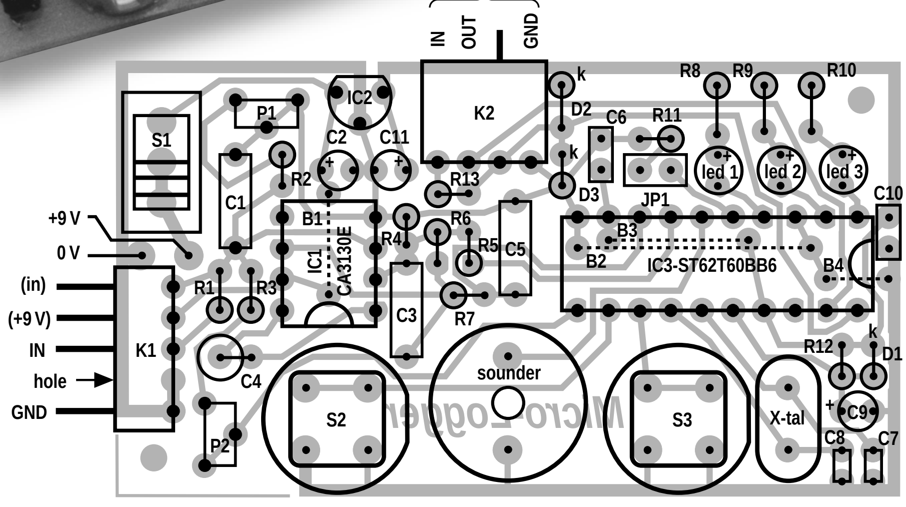
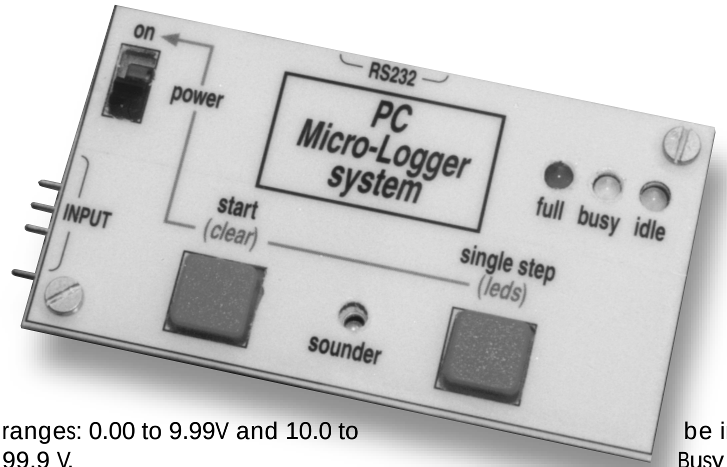


Figure 3. The circuit board layout (scale 1:1) and component layout of the micrologger (scale 2:1).



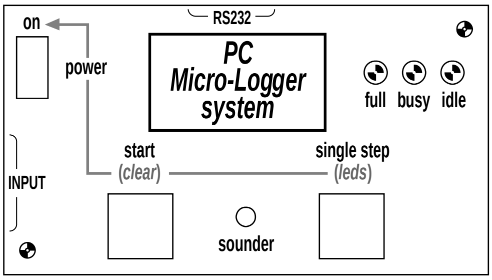


Figure 4. Sample front panel for the micrologger.

ranges: 0.00 to 9.99V and 10.0 to 99.9 V.

A pulse waveform at around 31 kHz from the ARTIM output is injected into the minus input of the opamp. Filter capacitors C3 and C5 are provided to extract the average value of this signal. The input of the micrologger is K1. In addition to the usual IN and GND inputs, a direct input (in) and the 9 V supply voltage are present at K1. The opamp circuit has a gain of around 9, and a voltage of around 2.2 V on the direct input results in an output voltage of 9.99 V. This means that an interface circuit can be connected to K1, for example for making current or temperature measurements.

The LEDs and S1, S2 and S3

The LEDs indicate the status of the micrologger. If no samples are being taken and the data memory is empty, the green LED will be on (Idle). If the memory contains sample data but is not full, the yellow LED will be on (Busy). When the memory is full, the red LED will be on (Full). The state of the LEDs is stored in the EEPROM. If the micrologger is switched on when the memory already contains sample data, this will be indicated by the Busy or Full LED being on. In this case, it is not possible to start sampling or to single step the micrologger. The memory must first be cleared. S1 is the on/off switch. If S2 is pressed while the Idle LED is on, the micrologger starts collecting data and no longer responds to S2 or S3. If S3 is pressed while the Idle LED is on, the micrologger takes a single sample of the actual voltage, and it no longer responds to S2. S3 may be pressed up to 82 times in succession.

If S2 is held pressed while the micrologger is switched on with S1, the memory is cleared. If S3 is held pressed while the micrologger is switched on with S1, the LEDs are switched on or off. Switching the LEDs off substantially reduces the current consumption of the micrologger. Although this setting is always saved in the EEPROM, there is one exception. If the LEDs have been switched off and samples are collected, the Full LED will be switched on when the memory is full.

Signals

The micrologger beeps after each sample has been taken, and after 82 samples — that is, when the memory is full — it beeps six times. If the memory is partially or completely filled and the micrologger is switched off and then switched on again, for example in order to read out the sample data to a PC, it will beep three times if you try to start sampling or to operate it in single-step mode (via the PC or using its push-buttons). The same thing happens if a 'step' command is issued via the PC while the micrologger is sampling, or vice versa. A new sampling or single-step session can be started only after the memory has been cleared.

If the micrologger is operated interactively with a PC, it beeps once if it receives an invalid input. This can be an incorrect command or an invalid parameter.

Building the micrologger

Figure 3 shows the printed circuit board and component layouts. Start with the four wire jumpers, and then mount the rest of the components. K1 is a male header with one more pin than is needed. To prevent incorrect mating of K1 with its matching connector, remove the 'extra' pin at the indicated

COMPONENTS LIST

Resistors:
(All resistors 0.25 or 0.2 W)
R1 = 499kΩ 1%
R2 = 127kΩ 1%
R3 = 10.7kΩ 1%
R4,R5 = 49.9kΩ 1%
R6 = 200kΩ 1%
R7,R13 = 10kΩ
R8,R9,R10 = 1kΩ
R11 = 4.7kΩ
R12 = 100kΩ
P1 = 25kΩ preset, 6 mm, vertical
P2 = 2.5kΩ preset, 6 mm, vertical

Capacitors:
C1 = 47nF, MKT, raster 7.5mm
C2,C11 = 4.7µF 15V, tantalum
C3 = 10nF, MKT, raster 7.5mm
C4 = 470pF, polystyrene (Siemens)
C5 = 3.3nF, MKT, raster 7.5mm
C6,C10 = 100nF, raster 0.1", multilayer
C7,C8 = 22pF, raster 0.1", ceramic
C9 = 1µF 15V, tantalum

Semiconductors:
IC1 = CA3130E
IC2 = LP2950CZ5.0
IC3 = ST62T60BB6 (programmed, order code 996518-1)
D1 = 1N4148
D2,D3 = BAT85
Led1 = LED 3 mm, high intensity, red
Led2 = LED 3 mm, high intensity, yellow
Led3 = LED 3 mm, high intensity, green

Miscellaneous:
S1 = 1-pole slide switch, 11 mm, PCB mount (Knitter type MFP 120)
S2,S3 = pushbutton, square (D6)
X-tal = quartz crystal 8 MHz, HC-49/S (height approx. 4 mm)
Snd = piëzo-ceramic buzzer, 14 mm diam., raster 7 mm
K1 = angled pinheader, SIL, 5 contacts
K1-contra = straight socket, SIL, 5 contacts
K2 = angled socket, , SIL, 4 contacts
K2-contra = straight pinheader, SIL, 4 contacts
JP1 = straight pinheader, 2 contacts
JP1-contra = jumper (shunt), 2.54 mm
B1-B4 = wire link
20-pin IC socket
Battery clip for 9-V PP3 battery
3.5"-floppy disk, contains project source-code: order code **996026-1**

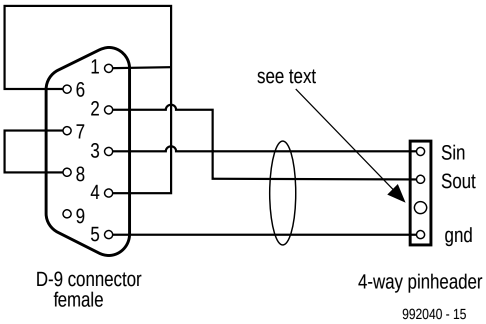


Figure 5. Cable for connecting the micrologger to the RS232 port of a PC. The D-type connector includes several cross connections to 'trick' the UART.

location and enlarge the hole using a 1.4-mm drill. Then glue a pin at the same location on the matching connector, using a small drop of crazy glue. Now the connectors can never be mated the wrong way around.

Treat K2 and its matching connector in the same manner. Finish off the leads of the matching connectors with heat-shrink tubing to give a neat appearance. Make sure that the flats of S2 and S3 face to the right.

Finally, you can copy the front panel layout in **Figure 4**, cut it out and glue it to a 1-mm thick piece of ABS plastic. After you have made the necessary openings, you can attach this panel to the circuit board using standoffs and 2-mm screws.

Software

Before the micrologger can be connected to the COM port of a PC, a cable must be constructed as shown in **Figure 5**. The 9-way plug can be connected either directly to the COM port or via a serial extension cable, with a 25-way connector at the PC end if necessary.

The HyperTerminal program can be used for communication with the PC. Start the program and select File/Properties. Set Connect to 'Direct to COMn', where n is number of the COM port used. Now choose Configure and enter: 38400, 7, none, 2, flow control = 'Hardware'. Select the Settings tab card and set Emulation = 'Auto detect'. Select ASCII Setup and make sure that only 'Wrap lines' is checked.

Save this file on the desktop as Logger.ht, or with any other suitable name (as long as it has the '.ht' extension). Close HyperTerminal and then start it up again, using the new icon on the desktop. Only then will the new settings take effect!

If the micrologger is not yet connected to the PC, connect it now and switch it on.

Tip: use only lower-case characters for communicating with the micrologger!

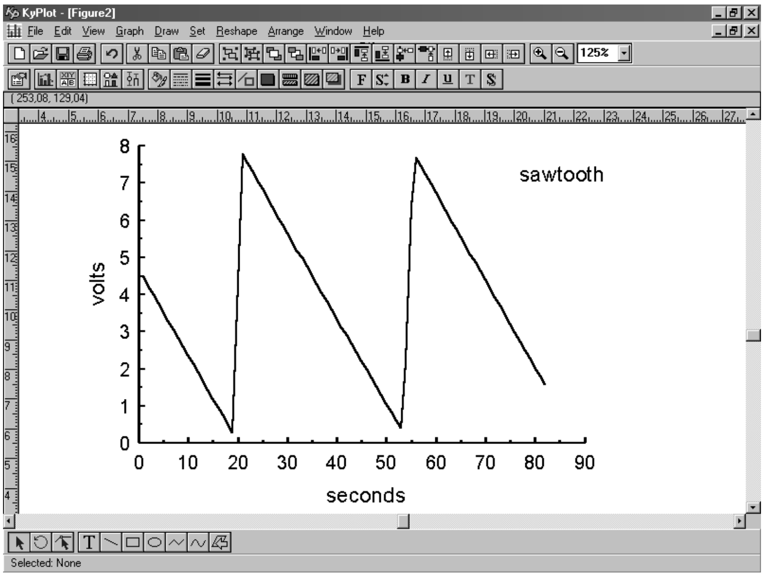


Figure 6. The file containing the measurement data can be loaded into KyPlot in order to directly create a graphic display of the data. Here you see a slow sawtooth waveform produced by a function generator.

Press Enter. The micrologger will beep and respond with 'command?'. Now press '?' and then Enter. The micrologger will respond with a list of all available commands.

The delay time can be set by entering it following the 'delay' command. To see what the current delay time setting is (and its format), simply type 'delay'.

The commands 'delim(iter)' and 'd(ecimal)p(oint)' change the formatting of the export file.

The remaining commands are self-explanatory, although there's one handy tip. If the 'volt' command is active, it can be switched off by pressing the Escape key. This can also be used to nullify any unintentionally entered command.

The sample data in the memory are exported in response to ' ^ d' (Control d). Before doing this, you can prepare a file to receive the data by selecting Transfer and then 'Capture text' in HyperTerminal, and giving the file a name with the extension '.txt'. The exported data will then be directly written to the specified file. Be sure to close this file before issuing any new commands. Alternatively, the samples can be selected on the screen and transferred via the clipboard to the notepad, for example.

The file data can be read directly into the KyPlot spreadsheet program [2], so that the data can be presented in graphic format.

Calibration

Individual examples of the microcontroller show slight differences in behaviour which, small as they may be, make it impossible to specify fixed values for the breakpoints at 25%, 50% and 75%. A simple but effective solution to this problem is found by including commands that allow these values to be adjusted; these are the 'inc' and 'dec' commands. You will need a good DVM and a variable power supply (0 to 15 V) to perform the calibration. If you have access to an oscilloscope, it's a good idea to connect it to the ARTIM output (pin 7). Using the oscilloscope, you can readily see where the breakpoints lie, and it is thus easier to adjust them for smooth transitions.

Connect both the DVM and the micrologger to the variable power supply, and adjust the voltage to approximately 2.00 V. Start HyperTerminal and enter the command 'volt'. Now adjust P1 until the reading on the screen is the same as that displayed by the DVM (tip: select a fairly large character size in

HyperTerminal, so it’s easy to read).
Now set the voltage to 4.00 V and use the commands ‘dec 0’ and ‘inc 0’ to adjust the value displayed on the screen until it is as close as possible to the DVM value. Decrease the voltage to around 2.4V, increase it very slowly to 2.6V and then reduce it again to 2.4V. The crossover at the breakpoint should be as smooth as possible.
Repeat this procedure with a voltage of 6.00 V and the commands ‘dec 1’ and ‘inc 1’ (breakpoint 5.0 V), and then with 8.00 V and the commands ‘dec 2’ and ‘inc 2’ (breakpoint 7.5 V).
Now adjust the voltage so the value displayed on the screen jumps between 9.99 V and some other value. Adjust P2 so that the ‘other value’ is 10.0 V (and not 09.9 V).
This completes the calibration process.

Conclusion

The accuracy of the clock should be more than adequate in most cases. It can be checked by setting the delay time to one hour. Wait until the seconds hand of your watch reaches 12, and start the micrologger. It will emit a beep then, and every hour thereafter, up to 82 times. Each time a beep occurs, the seconds hand of your watch will indicate the extent to which the micrologger clock runs fast or slow. If necessary, the clock rate can be adjusted by experimenting with the values of C7 and C8.
The SPI is used for both reception and transmission. When the ‘volt’ command is active, it can happen that SPI is transmitting at exactly the same time as a key is pressed, in which case the key code will be lost. If this happens, press Enter and then enter the command once again.

(990062)

Literature

[1] ST Application Note AN420/0294, ‘Expanding A/D Resolution of the ST6 A/D converter’.

[2] KyPlot http://www.qualest.co.jp/Download/KyPlot/kyplot_e.htm (see Elektor Electronics March 1999 issue).